International Journal of Computational Intelligence in Control
Vol. 12 No. 2 (July-December, 2020)

BACKSTEPPING-BASED GENETIC FUZZY-NEURAL CONTROLLER AND ITS APPLICATION IN MOTOR CONTROL WITH BUCK DC-DC CONVERTERS

Yih-Guang Leu, Jian-You Lin and Yi-Chuan Lu
Department of Applied Electronics Technology, National Taiwan Normal University
162, He-Ping E. Road, Sec 1, Taipei, Taiwan

***ABSTRACT*:** In this paper, a genetic adaptive fuzzy-neural control scheme is proposed for a class of multiple-input multiple-output (MIMO) nonlinear systems. The control scheme incorporates backstepping design into the genetic algorithm with a backstepping-based fitness function. Using the backstepping-based fitness function, the genetic algorithm can be used to adjust the parameters of the fuzzy-neural networks in order to instantaneously generate the appropriate control strategy. The genetic algorithm has a simplified procedure with the backstepping-based fitness function which is used to evaluate the real-time stability of the closed-loop systems. To illustrate the feasibility and applicability of the proposed method, simulation and experimental results are provided.

Keywords: Genetic algorithm, adaptive fuzzy-neural control, nonlinear systems

1. INTRODUCTION

The design of fuzzy logic systems and/or neural networks for adaptive controllers [1-3] has been widely developed because of the universal approximation feature [4-5], and the stability analysis of the adaptive fuzzy logic and/or neural network controllers for nonlinear systems is generally provided by Lyapunov stability theory. To search global optimal solutions, genetic algorithms [6-23] have been incorporated into the design of fuzzy logic systems and/or neural networks systematically because they are based on natural selection and natural genetics, and possess the simple implement ability and the capability of escaping from local optima. Motor servo control via genetic algorithms has been proposed in [9]. In [20-22], for the fuzzy-neural networks, the learning process utilizes genetic algorithms rather than the conventional learning methods. In [23], to reduce the computation loading, a reduced-form genetic algorithm has been proposed for function approximation.

For the design of adaptive controllers of nonlinear systems, the complicated mathematical form for fuzzy logic systems and/or neural networks [1-3], such as the update laws and the Lyapunov condition for the system stability, must be solved. For this reason, it is difficult to implement the control algorithms into real adaptive controllers. Moreover, the design of adaptive controllers incorporated into genetic algorithms generally requires the procedure of off-line learning [7-9] before they on-line control a plant. Thus, in this paper, to avoid solving complicated mathematical equations, a genetic algorithm controller without the procedure of off-line learning is developed for nonlinear systems, and the stability of the closed-loop system is guaranteed. Also, to avoid the cancellation of useful nonlinearities in the design process for nonlinear systems, adaptive backstepping control technique [24] is used. More specifically, we propose an adaptive backstepping fuzzy-neural controller using the genetic algorithm with the backstepping-based fitness function for a class of multiple-input multiple-output (MIMO) nonlinear systems. The weighting factors of the adaptive fuzzy-neural controller are tuned on-line via the genetic algorithm, instead of solving complicated mathematical equations. For the purpose of on-line

Corresponding author: E-mail: leuyk@ntnu.edu.tw

tuning these parameters and evaluating the stability of the closed-loop system, the backstepping-based fitness function is included in the genetic algorithm.

2. PROBLEM FORMULATION AND FUZZY-NEURAL NETWORKS

First, consider the MIMO nonlinear systems as

$$\begin{aligned}
\dot{x}_{p1} &= x_{p2} \\
\dot{x}_{p2} &= x_{p3} \\
&\vdots \\
\dot{x}_{pn_p} &= f_p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_h) + b_p u_p
\end{aligned} \quad , p = 1, 2, \ldots, m \tag{1}$$

where f_p is the unknown system dynamics of the p-th subsystem, u_p is the input of the p-th subsystem, b_p is a positive constant, $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_m]^T$ is the state vector, and $\mathbf{x}_p = [x_{p1}, x_{p2}, \ldots x_{pn_p}]^T$ is the state vector of the p-th subsystem. Our control objective is to develop the backstepping controller so that the state trajectory x_{p1} can asymptotically track a bounded command y_{pd}.

Next, the detail design procedure of the backstepping controller under the assumption of the known system dynamics f_p is described as follows.

Step 1: Define a tracking error as

$$z_{p1} = x_{p1} - y_{pd} \tag{2}$$

Then, differentiating z_{p1} can be expressed as

$$\dot{z}_{p1} = \dot{x}_{p1} - \dot{y}_{pd} \tag{3}$$

Define a virtual control as

$$\alpha_{p1} = \dot{y}_{pd} - c_{p1} z_{p1} \tag{4}$$

where $c_{p1} > 0$ is a design parameter. From (3) and (4), if $\alpha_{p1} = \dot{x}_{p1}$, then $\lim_{t\to\infty} z_{p1} \to 0$, that is, the state trajectory x_{p1} can asymptotically track the bounded command y_{pd}. Thus, define an error state as $z_{p2} = \dot{x}_{p1} - \alpha_{p1} = x_{p2} - \alpha_{p1}$. Then, our next goal is to force the error state z_{p2} to decay to zero. By using (4) and the fact that $\dot{x}_{p1} = z_{p2} + \alpha_{p1}$, equation (3) can be rewritten as

$$\dot{z}_{p1} = z_{p2} - c_{p1} z_{p1} \tag{5}$$

Step 2: Differentiating z_{p2} can be expressed as

$$\dot{z}_{p2} = \dot{x}_{p2} - \dot{\alpha}_{p1} = x_{p3} - (-c_{p1}\dot{z}_{p1} + \ddot{y}_{pd}) \tag{6}$$

Similarly, define a virtual control as

$$\alpha_{p2} = \ddot{y}_{pd} - c_{p1}\dot{z}_{p1} - c_{p2} z_{p2} - z_{p1} \tag{7}$$

where $c_{p2} > 0$ is a design parameter. Moreover, define an error state as $z_{p3} = x_{p3} - \alpha_{p2}$. Then, by using (7) and the fact that $\dot{x}_{p2} = z_{p3} + \alpha_{p2}$, equation (6) can be rewritten as

$$\dot{z}_{p2} = z_{p3} - c_{p2} z_{p2} - z_{p1} \tag{8}$$

Step 3: Let k be a positive integer. Define an error state as $z_{pk} = x_{pk} - \alpha_{p(k-1)}$. Then, differentiating z_{pk}, where $3 \le k \le n_p - 1$, can be expressed as

$$\dot{z}_{pk} = \dot{x}_{pk} - \dot{\alpha}_{p(k-1)} \tag{9}$$

Define a virtual control as

$$\alpha_{pk} = y_{pd}^{(k)} - \sum_{i=1}^{k} c_{pi} z_{pi}^{(k-i)} - \sum_{j=1}^{k-1} z_{pj}^{(k-1-j)} \tag{10}$$

where $c_{pi} > 0$ is a design parameter. Moreover, define an error state as $z_{p(k+1)} = x_{p(k+1)} - \alpha_{pk}$. Then, by using (10) and the fact that $\dot{x}_{pk} = z_{p(k+1)} + \alpha_{pk}$, equation (9) can be rewritten as

$$\dot{z}_{pk} = z_{p(k+1)} - c_{pk} z_{pk} - z_{p(k-1)} \tag{11}$$

Step 4: Differentiating z_{pn_p} can be expressed as

$$\dot{z}_{pn_p} = \dot{x}_{pn_p} - \dot{\alpha}_{p(n_p-1)} = f_p + b_p u_p - \left(y_{pd}^{(n_p)} - \sum_{i=1}^{n_p-1} c_{pi} z_{pi}^{(n_p-i)} - \sum_{j=1}^{n_p-2} z_{pj}^{(n_p-1-j)} \right) \tag{12}$$

Define a control law as

$$u_p = \frac{1}{b_p} \left(-f_p + y_{pd}^{(n_p)} - \sum_{i=1}^{n_p} c_{pi} z_{pi}^{(n_p-i)} - \sum_{j=1}^{n_p-1} z_{pj}^{(n_p-1-j)} \right) \tag{13}$$

where $c_{pn_p} > 0$ is a design parameter. Then, from (13), equation (12) can be rewritten as

$$\dot{z}_{pn_p} = -c_{pn_p} z_{pn_p} - z_{p(n_p-1)} \tag{14}$$

Step 5: Consider the Lyapunov function as follows

$$V = \frac{1}{2} \sum_{p=1}^{m} \sum_{i=1}^{n_p} z_{pi}^2 \tag{15}$$

By differentiating (15) and using (5), (8), (11) and (14), we have

$$\begin{aligned} \dot{V} &= \sum_{p=1}^{m} \sum_{i=1}^{n_p} z_{pi} \dot{z}_{pi} = -\sum_{p=1}^{m} \sum_{i=1}^{n_p} c_{pi} z_{pi}^2 \\ &\leq -\sum_{p=1}^{m} c_{p1} z_{p1}^2 \end{aligned} \tag{16}$$

From (15) and (16), we can conclude that z_{pi} is bounded. Moreover, from (5), $\dot{z}_{p1}$ is also bounded. Integrating (16) can be expressed as

$$\sum_{p=1}^{m} c_{p1} \int_0^{\infty} z_{p1}^2(\tau) d\tau \leq -V(\infty) + V(0) \tag{17}$$

Because of the fact that the right side of (17) is bounded, we have $z_{p1} \in L_2$. According to the Barbalat's Lemma [25], $\lim_{t\to\infty} z_{p1} = 0$, that is, the state trajectory x_{p1} can asymptotically track the bounded command y_{pd}.

On the basis of the aforementioned description, the backstepping controller of the nth order nonlinear system can be summarized as the following lemma.

Lemma: Consider the nth-order nonlinear systems (1). Let $z_{p1}=x_{p1}-y_{pd}$ and $z_{pi}=x_{pi}-\alpha_{p(i-1)}$ for $2\le i\le n_p$, where $\alpha_{p1}=\dot{y}_{pd}-c_{p1}z_{p1}$, and $\alpha_{pk}=y_{pd}^{(k)}-\sum_{i=1}^{k}c_{pi}z_{pi}^{(k-i)}-\sum_{j=1}^{k-1}z_{pj}^{(k-1-j)}$ for $2\le k\le n_p-1$. Suppose that the control law is given as $u_p=\frac{1}{b_p}\left(-f_p+y_{pd}^{(n_p)}-\sum_{i=1}^{n_p}c_{pi}z_{pi}^{(n_p-i)}-\sum_{j=1}^{n_p-1}z_{pj}^{(n_p-1-j)}\right)$, where $c_{pi}>0$. Then, the state trajectory x_{p1} can asymptotically track the bounded command y_{pd}.

The fuzzy-neural network architecture in [2] is used in this paper. The fuzzy inference engine uses the fuzzy IF-THEN rules to perform a mapping from an training input data x_{pq}, $p = 1, 2, ..., m$, $q = 1, 2, ..., n_p$, and the output data y_p, $p = 1, 2, ..., m$, the ith fuzzy rule has the following form:

$$\begin{aligned} R^i: IF \quad & x_{11} \quad is \quad A_{11}^i \quad and \cdots x_{mn_m} \quad is \quad A_{mn_m}^i \\ THEN \quad & y_1 \quad is \quad B_1^i \quad and \cdots y_m \quad is \quad B_m^i \end{aligned} \tag{18}$$

where i is a rule number, A_{pq}^i and B_p^i are the fuzzy sets. By using product inference, center-average and singleton fuzzifier, the output of the fuzzy-neural networks can be expressed as:

$$y_p(\mathbf{x}|w_p)=\frac{\sum_{i=1}^{N}w_p^i\left(\prod_{s=1}^{m}\prod_{q=1}^{n_s}\mu_{A_{sq}^i}(x_{sq})\right)}{\sum_{i=1}^{N}\left(\prod_{s=1}^{m}\prod_{q=1}^{n_s}\mu_{A_{sq}^i}(x_{sq})\right)}=\mathbf{w}_p^T\varphi(\mathbf{x}) \tag{19}$$

where $\mu_{A_{sq}^i}(x_{sq})$ is the membership function of A_{sq}^i, N is the total number of the IF-THEN rules, w_p^i is the point at which $\mu_{B^i}\left(w_p^i\right)=1$, $\mathbf{w}_p=[w_p^1\ w_p^2\cdots w_p^N]^T$ is a weighting vector, $\varphi^T=[\varphi^1\ \varphi^2\cdots\varphi^N]$ is the fuzzy basis vector, where φ^i is defined as

$$\varphi^i(\mathbf{x})=\frac{\prod_{s=1}^{m}\prod_{q=1}^{n_s}\mu_{A_{sq}^i}(x_{sq})}{\sum_{i=1}^{N}\left(\prod_{s=1}^{m}\prod_{q=1}^{n_s}\mu_{A_{sq}^i}(x_{sq})\right)} \tag{20}$$

3. DESCRIPTION OF GENETIC ALGORITHM

For the purpose of speeding up the computation of the genetic operation [23], the mechanism of the genetic algorithm has three simplified parts: (1) keep a small population size, (2) replace the string codes with real-value representation, and (3) perform compact mutation and crossover on the chromosomes by the backstepping-based fitness function. The details are discussed in the following.

First, for the adjustable parameters of the p-th output of the fuzzy-neural networks, define a population of solutions with ξ chromosomes as

$$\Xi_p=\begin{bmatrix} {}^1\mathbf{w}_p^T \\ {}^2\mathbf{w}_p^T \\ \vdots \\ {}^\xi\mathbf{w}_p^T \end{bmatrix} \tag{21}$$

where ${}^{i}\mathbf{w}_p$ denotes the ith chromosome and is a set of weighting factors in the interval $\mathbf{D}_p = [-d_p, d_p], d_p > 0$. Each gene represents the adjustable parameter of the fuzzy-neural networks. Note that the population is sorted by ranking the fitness of chromosomes, that is, the first chromosome denotes the best chromosome in the population in terms of fitness.

Next, to instantaneously evaluate the stability of the closed-loop system, define the p-th backstepping-based fitness function for the p-th subsystem as

$$F_p = z_{pn_p} \hat{f}_p(\mathbf{x} \mid \mathbf{w}_p) \tag{22}$$

where $\hat{f}_p(\mathbf{x} \mid \mathbf{w}_p)$ is the estimation of the unknown dynamic $f_p(\mathbf{x} \mid \mathbf{w}_p)$. A chromosome with largest fitness denotes the optimal solution. The detail explanation of the backstepping-based fitness function is given later.

Then, according to the backstepping-based fitness function, mutation and crossover operators are performed. The operation procedure of the crossover operator is as follows. Only a single gene is randomly chosen to perform the crossover operation. Suppose that the j-th gene is selected as the crossover point. Then, the j-th gene of all chromosomes is updated as

$$ {}^{i}w_p^j = \begin{cases} {}^{i}w_p^j * {}^{i}\zeta_p + {}^{i+(\xi/2)}w_p^j * (1 - {}^{i}\zeta_p) & \text{for } i = 1, 2, \ldots, \xi/2 \\ {}^{i}w_p^j * {}^{i}\kappa_p + {}^{i-(\xi/2)}w_p^j * (1 - {}^{i}\kappa_p) & \text{for } i = (\xi/2)+1, (\xi/2)+2, \ldots, \xi \end{cases} \tag{23}$$

where the crossover factors ${}^{i}\zeta_p$ and ${}^{i}\kappa_p$ are the combination weights between the two crossover chromosomes according to the backstepping-based fitness function, and they are defined as

$${}^{i}\zeta_p = \frac{{}^{i}F_p - {}^{\xi}F_p}{[{}^{i}F_p - {}^{\xi}F_p] + [{}^{i+(\xi/2)}F_p - {}^{\xi}F_p]} \tag{24}$$

$${}^{i}\kappa_p = \frac{{}^{i}F_p - {}^{\xi}F_p}{[{}^{i}F_p - {}^{\xi}F_p] + [{}^{i-(\xi/2)}F_p - {}^{\xi}F_p]} \tag{25}$$

respectively.

As for the operation procedure of the mutation operator, the $(\xi/2{+}1)$-th chromosome is replaced by the first chromosome according to mutation rate p_m. Then, the $(\xi / 2{+}1)$-th chromosome is updated as

$$ {}^{(\xi/2+1)}w_p^j = \begin{cases} {}^{1}w_p^j + \gamma^{\beta} * \eta_p, & \text{if } \delta > 0.5 \\ {}^{1}w_p^j, & \text{if } \delta \le 0.5 \end{cases} \tag{26}$$

where

$$\eta_p = \begin{cases} d_p - {}^{1}w_p^j, & \text{if } \lambda > 0.5 \\ -d_p - {}^{1}w_p^j, & \text{if } \lambda \le 0.5 \end{cases} \tag{27}$$

and $\delta, \gamma, \lambda \in [0, 1]$ are random numbers, and $\beta > 0$ is a design parameter.

4. DEVELOPMENT OF GENETIC ADAPTIVE FUZZY-NEURAL CONTROL SCHEME

Since f_p are uncertain, the optimal control law (13) cannot be obtained. To solve this problem, the fuzzy-neural system is used to approximate the uncertain continuous function f_p. First, the uncertain continuous function f_p in (13) is replaced by fuzzy-neural networks (19), i.e., $\hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p)$. The resulting control law

$$u_{pc} = \frac{1}{b_p}\left(y_d^{(n_p)} - \sum_{i=1}^{n_p} c_{pi} z_{pi}^{(n_p-i)} - \sum_{j=1}^{n_p-1} z_{pj}^{(n_p-1-j)} - \hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p)\right) \tag{28}$$

Suppose that $u_p = u_{pc}$. Then, substituting (28) in (1) and after some manipulations, we obtain the error equation

$$\dot{z}_{pn_p} = f_p(\mathbf{x}) - \hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p) - c_{pn_p} z_{pn_p} - z_{p(n_p-1)} \tag{29}$$

Let $V = \frac{1}{2}\sum_{p=1}^{m}\sum_{i=1}^{n_p} z_{pi}^2$. By using (29), we have

$$\begin{aligned}
\dot{V} &= \sum_{p=1}^{m}\sum_{i=1}^{n_p} z_{pi}\dot{z}_{pi} \\
&= \sum_{p=1}^{m}[z_{p1}(z_{p2} - c_{p1}z_{p1}) + \sum_{i=2}^{n_p-1} z_{pi}(z_{p(i+1)} - c_{pi}z_{pi} - z_{p(i-1)}) \\
&\quad + z_{pn_p}(f_p(\mathbf{x}) - \hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p) - c_{pn_p}z_{pn_p} - z_{p(n_p-1)})] \\
&= \sum_{p=1}^{m}[\sum_{i=1}^{n_p} -c_{pi}z_{pi}^2 + z_{pn_p}(f_p(\mathbf{x}) - \hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p))] \\
&\le \sum_{p=1}^{m}[-\sum_{i=1}^{n_p} c_{pi}z_{pi}^2 + \left|z_{pn_p}\right| f_p^{\,U}(\mathbf{x}) - z_{pn_p}\hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p)]
\end{aligned} \tag{30}$$

where $\left|f_p(\mathbf{x})\right| \le f_p^{\,U}(\mathbf{x}) < \infty$. By using the fact that the states must move in the direction of smaller values of V if $\dot{V} < 0$, the p-th backstepping-based fitness function $z_{pn_p}\hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p)$ in (22) for the p-th subsystem is defined in order to instantaneously evaluate the stability of the closed-loop system. A chromosome with the largest backstepping-based fitness function denotes the optimal solution. So, a better chromosome can be obtained according to the backstepping-based fitness function (22).

Since the system states may go into the unsafe region if the genetic operations can not simultaneously generate the appropriate weightings of the fuzzy-neural networks in some time interval, the concept of the safe controller is incorporated into the genetic adaptive backstepping fuzzy-neural controller to guarantee that the system states are confined to the safe region. By incorporating a safe control term u_{ps} into u_{pc}, the control law becomes

$$u_p = u_{pc} + u_{ps} \tag{31}$$

The safe control term u_{ps} is added when the function V is greater than a positive limit V^u. If $V \le V^u$, then the safe control term u_{ps} is canceled. That is, if the system tends to enter the unsafe region ($V > V^u$), then u_{ps} forces the system to return to the safe region.

Substituting (31) into (1), the error equation becomes

$$\dot{z}_{pn_p} = f_p(\mathbf{x}) - \hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p) - c_{pn_p} z_{pn_p} - z_{p(n_p-1)} + b_p u_{ps} \tag{32}$$

Using (31) and (32), we have

$$\begin{aligned}\dot{V} &= \sum_{p=1}^{m}[-\sum_{i=1}^{n_p} c_{pi} z_{pi}^2 + z_{pn_p}(f_p(\mathbf{x}) - \hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p) + b_p u_{ps})] \\ &\le \sum_{p=1}^{m}[-\sum_{i=1}^{n_p} c_{pi} z_{pi}^2 + \left|z_{pn_p}\right|(\left|f_p(\mathbf{x})\right| + \left|\hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p)\right|) + z_{pn_p} b_p u_{ps}]\end{aligned} \tag{33}$$

Suppose that the safe control term u_{ps}[1] is given as

$$u_{ps} = \begin{cases} -\mathrm{sgn}(z_{pn})[f_p^{\,U}(\mathbf{x}) + \left|\hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p)\right|]/b_p & \text{if } V > V^u \\ 0 & \text{if } V \le V^u \end{cases} \tag{34}$$

where V^u is a design parameter and $\mathrm{sgn}(z_{pn})$ = 1(–1) if $z_{pn} \ge 0(<0)$. Suppose that $V>V^u$. Then, substituting (34) into (33), we have

$$\begin{aligned}\dot{V} &\le \sum_{p=1}^{m}[-\sum_{i=1}^{n_p} c_{pi} z_{pi}^2 + \left|z_{pn_p}\right|[\left|f_p(\mathbf{x})\right| + \left|\hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p)\right| - f_p^{\,U}(\mathbf{x}) - \left|\hat{f}_p(\mathbf{x}\,|\,\mathbf{w}_p)\right|]] \\ &\le -\sum_{p=1}^{m}\sum_{i=1}^{n_p} c_{pi} z_{pi}^2 \le 0\end{aligned} \tag{35}$$

From (35), the bounded stability of the closed-loop system for the nonlinear system in (1) can be guaranteed.

5. SIMULATION EXAMPLES

Consider the following problem of balancing double inverted pendulums connected by a spring [26]

$$\begin{aligned}\dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= \frac{m_1 g r}{J_1}\sin(x_{11}) - \frac{k}{J_1}x_{11} + \frac{u_1}{J_1} + \frac{k}{J_1}x_{21} \\ \dot{x}_{21} &= x_{22} \\ \dot{x}_{22} &= \frac{m_2 g r}{J_2}\sin(x_{21}) - \frac{k}{J_2}x_{21} + \frac{u_2}{J_2} + \frac{k}{J_2}x_{11}\end{aligned} \tag{36}$$

where x_{i1} is the angular position of the ith pendulum from the vertical reference, and u_i is a torque input. It is assumed that both x_{i1} and $\dot{x}_{i1}$ are available for measurement. The parameters of the double inverted pendulums are chosen as m_1 = 0.5 kg m_2 = 0.5 kg, J_1 = 0.5 kg and J_2 = 0.5 kg, k = 2N $\cdot$ m/rad, and r = 1 m.

Our objective is to control the system state x_{p1} to track the reference trajectory y_{pd}. The design parameters of the genetic algorithm are given as ξ = 4, p_m = 0.06, and β = 4. The adjustable parameters w_i of $\hat{f}_i(x_{11}, x_{12}, x_{21}, x_{22})$ are in the intervals D_i = [–1,1]. The reference signals are set as $y_{1d}(t) = (2.5\pi/12)\sin(t)$ and $y_{2d}(t) = (3.75\pi/12)\cos(t)$. The membership functions for x_{pi} are given as $\mu_{A_i^1}(x_{pi}) = e^{-(x+1)^2}$, $\mu_{A_i^2}(x_{pi}) = e^{-(x+0.5)^2}$, $\mu_{A_i^3}(x_{pi}) = e^{-x^2}$, $\mu_{A_i^4}(x_{pi}) = e^{-(x-0.5)^2}$, and $\mu_{A_i^5}(x_{pi}) = e^{-(x-1)^2}$.

The initial states are set as $x(0) = [0.5, 0, 0.3, 0]$. The design parameters are selected as $c_{11} = 10$, $c_{12} = 10$, $c_{21} = 8$, and $c_{22} = 8$. The simulation results are shown in Figs. 1-4. Figs. 1-4 show that the system outputs x_{11} and x_{21} can track the reference signals $y_{1d}(t)$ and $y_{2d}(t)$ very well, respectively.

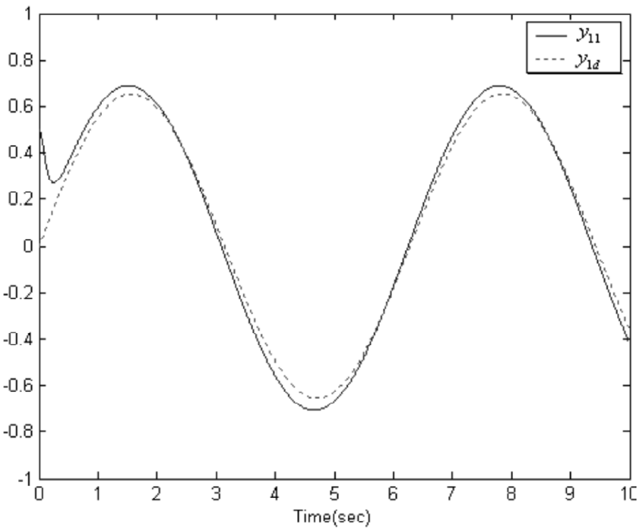

Figure 1: The System Output $y_{11}(t)$ and Bounded Reference $y_{1d}(t)$

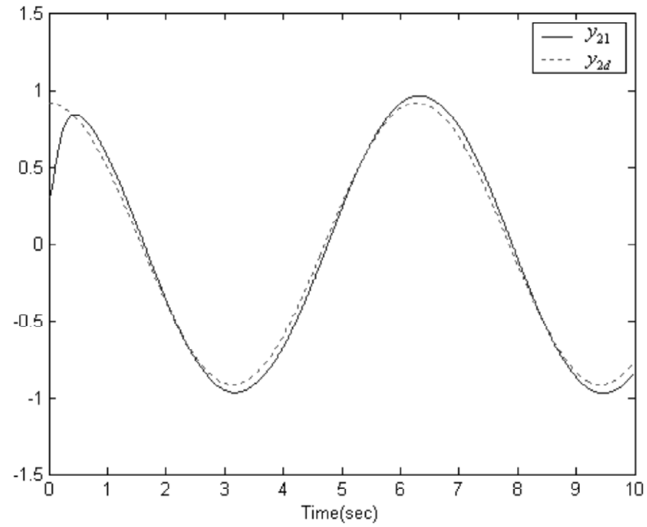

Figure 2: The System Output $y_{21}(t)$ and bounded reference $y_{2d}(t)$

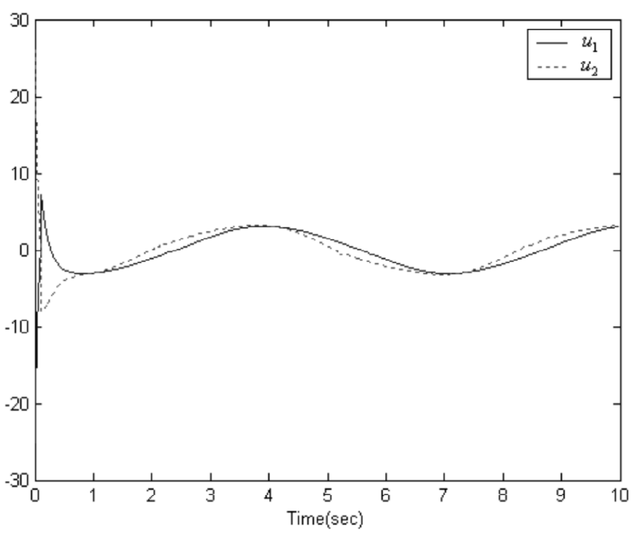

Figure 3: The Control Inputs u_1 and u_2

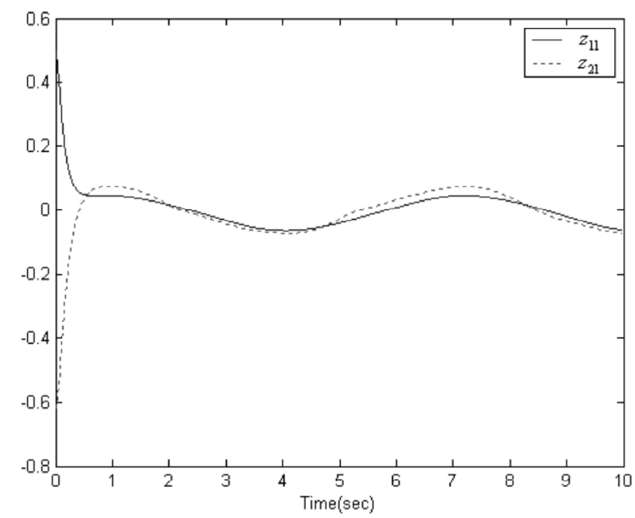

Figure 4: The Tracking Errors $z_{11}(t)$ and $z_{21}(t)$

6. EXPERIMENTAL RESULTS

Consider the servo motor described as

$$\frac{\theta(s)}{V_a(s)} = \frac{K_t}{s\left[JLs^2 + (JR + BL)s + (BR + K_t K_b)\right]} \tag{37}$$

where R is the armature resistance, L is the armature inductance, θ is the angular displacement of the motor shaft, B is the friction constant, J is the armature moment of inertia of the motor, and K_b is a voltage constant, K_t is a torque constant, and V_a is the armature voltage. In general, the inductance L is small and may be neglected. If L is neglected, then the state space of transfer function (37) can be expressed as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -(K_t K_b + RB)/JR \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ K_t / JR \end{bmatrix} u \tag{38}$$

where $\mathbf{x}(t) = [x_1(t), x_2(t)]^T = [\theta(t)\ \dot{\theta}(t)]^T$ and $u = v_a$. The proposed method controls the SANYO V511-612EL8 DC servo motor with $K_t = 0.21\text{Nm/A}$, $R = 4.8\Omega$, $K_b = 0.21\text{volts}\cdot\text{sec/rad}$, $J = 0.037\times10^{-3}\text{Kgm}^2$ and $B = 0.013\text{N-m-s/rad}$. The control objective is to force the speed $x_2 = w$ of the servo motor to follow the reference speed w_r.

The control system block diagram is shown in Fig. 5. The servo driver motor system is controlled by PWM (Pulse-Width Modulation) method where switch-duty ratio $d \in [0, 1]$ is varied to adjust the output of the Buck DC-DC converter. A proportional–integral–derivative (PID) controller in inner loop is used to control the output voltage of the Buck DC-DC converter through the switch duty ratio d, and then the proposed method in outer loop is used to control the speed of the servo driver motor. The design parameters of the genetic algorithm are given as $k = 4$, $p_m = 0.06$, and $\beta = 4$. The control parameters are selected as $c_1 = 100$, $c_2 = 100$, $V^u = 0.1$. The PID gains are given as $K_p = 25$, $K_I = 65$, and $K_D = 75$.

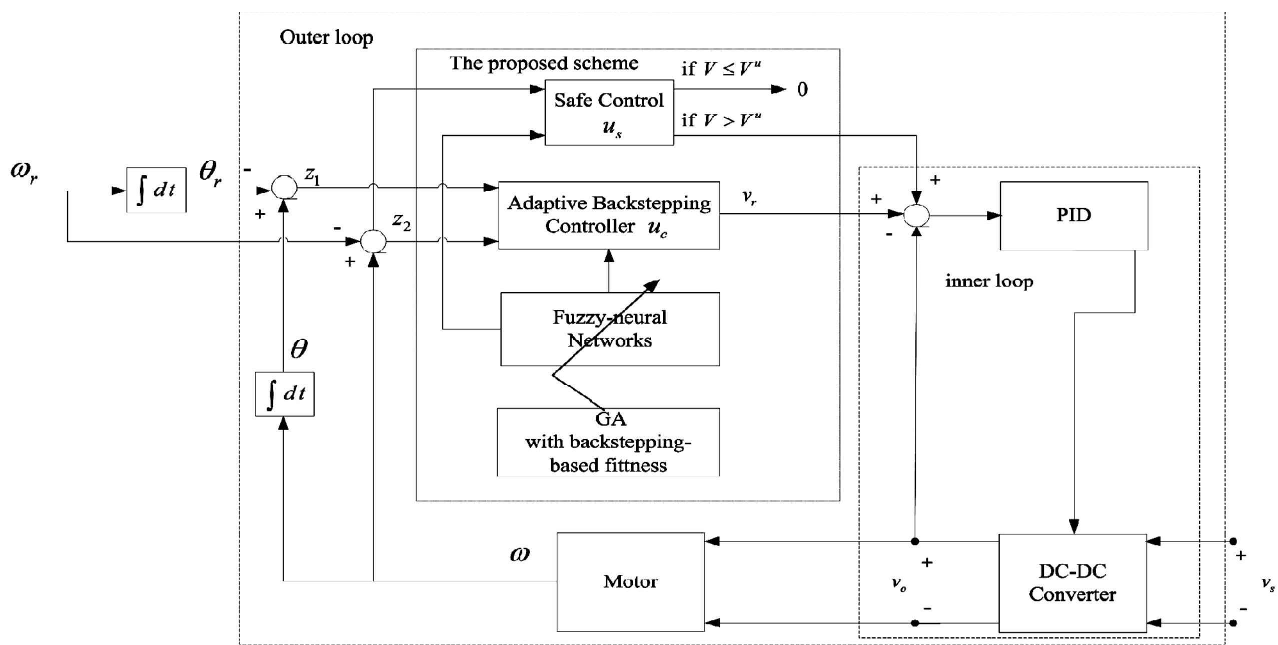

Figure 5: Block Diagram of the Motor System with DC-DC Converter

Traditionally, using two PID controllers in inner and outer loops controls the motor system with the DC-DC converter, respectively. To verify the effectiveness of the proposed method, we compare the proposed scheme with the conventional PID method.

The adjustable parameters $\mathbf{w}$ of $\hat{f}$ are in the intervals $D = [-5,5]$. The membership functions for x_i are given as $\mu_{A_i^1}(x_i) = 1/(1+e^{0.5\times(x_i+35)})$, $\mu_{A_i^2}(x_i) = e^{-0.001\times(x_i+15)^2}$, $\mu_{A_i^3}(x_i) = e^{-0.001\times(x_i+5)^2}$, $\mu_{A_i^4}(x_i) = e^{-0.001x_i^2}$, $\mu_{A_i^5}(x_i) = e^{-0.001\times(x_i-5)^2}$, $\mu_{A_i^6}(x_i) = e^{-0.001\times(x_i-15)^2}$, and $\mu_{A_i^7}(x_i) = 1/(1+e^{-0.5\times(x_i-35)})$.

For a constant reference ω_r = 30rad/s and a source voltage v_s = 10V, Figs. 6-9 show the experimental results of the proposed method and the conventional PID method, respectively. In both methods, the controlled system in Fig. 5 adds a shunt load resistance R_L = 20Ω in the output of the Buck DC-DC converter at 3-4 sec and 7-8 sec. It is shown that the motor velocity tracks its reference well in Figs.6 and 8 even though the controlled system has the load uncertainty. The mean square errors (MSE) of the tracking velocity error for the proposed method and the conventional PID method are equal to 6.917×10^{-2} and 8.445×10^{-2}, respectively. According to the accumulation of the tracking velocity error shown in Figs. 7 and 9, the proposed method is quite satisfactory as compared with the conventional PID method.

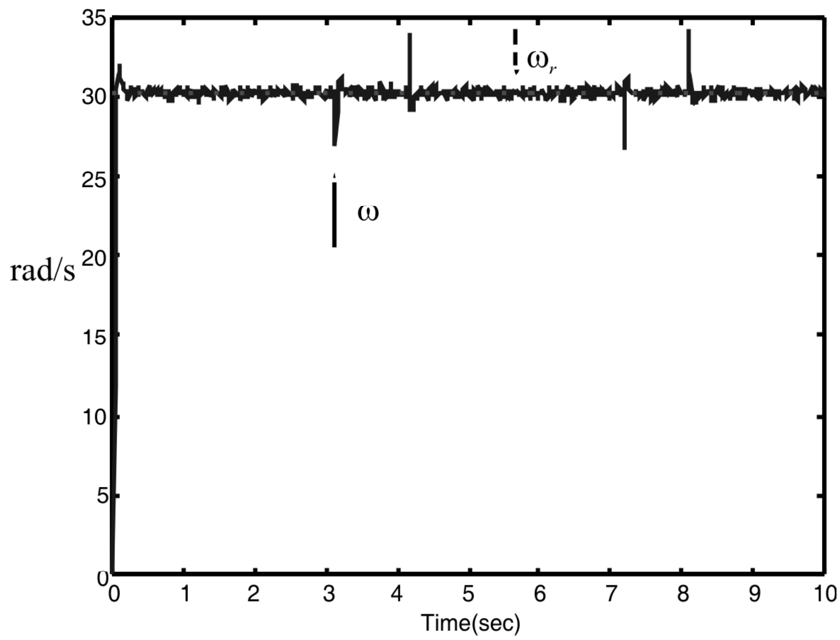

Figure 6: The System Output ω(t) and Bounded Reference $\omega_r(t)$ using the Proposed Method

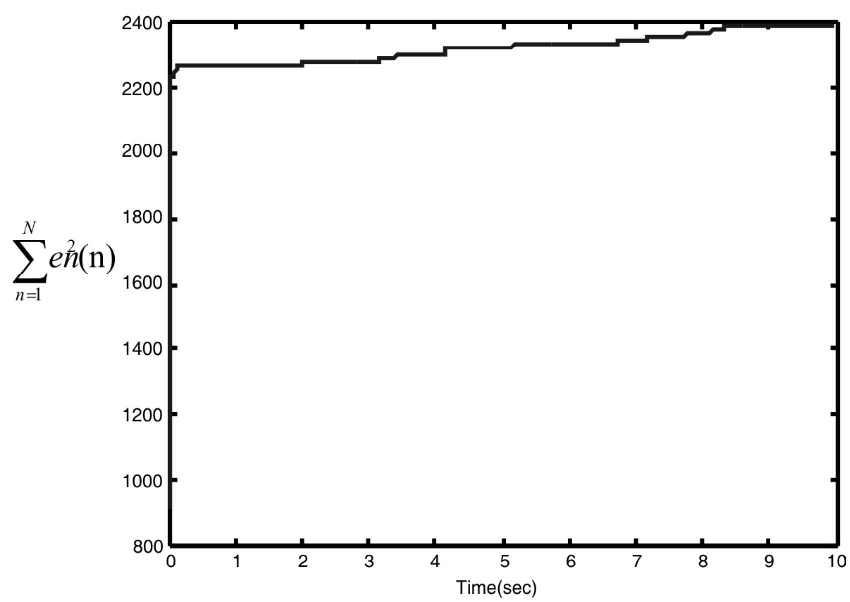

Figure 7: The Accumulation of the Tracking Error $e = w_r - w$ using the Proposed Method

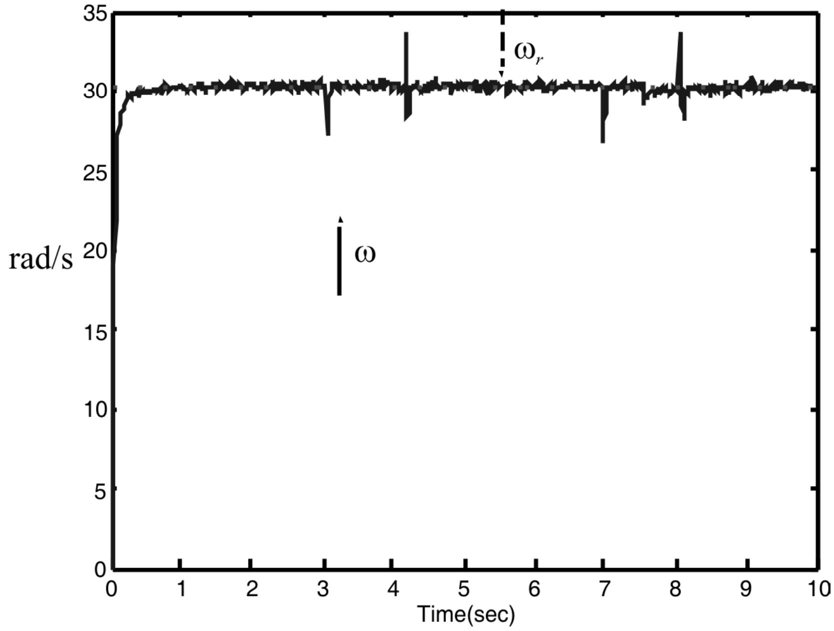

Figure 8: The System Output $\omega(t)$ and bounded Reference $\omega_r(t)$ using Conventional PID Method

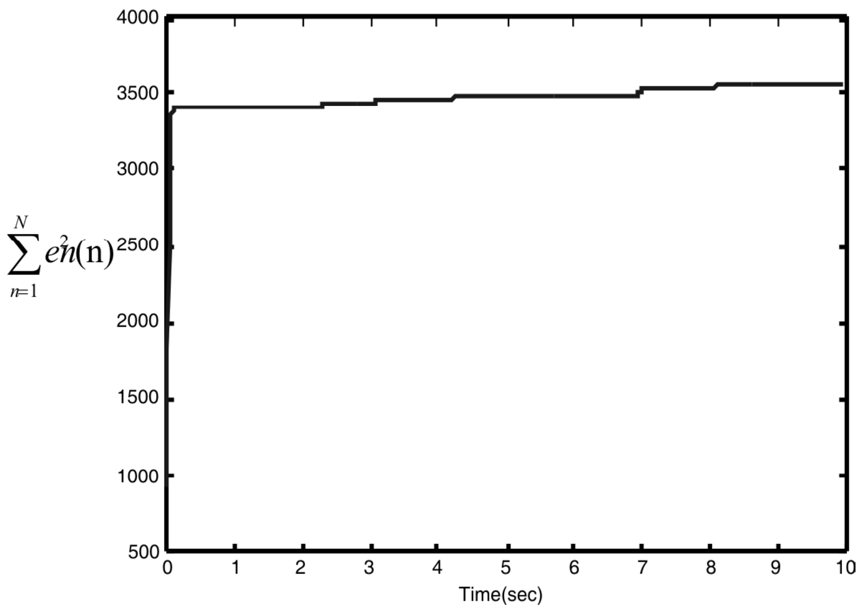

Figure 9: The Accumulation of the Tracking Error $e = w_r - w$ using Conventional PID Method

7. CONCLUSIONS

In this paper, a genetic adaptive fuzzy-neural control scheme has proposed for a class of MIMO nonlinear systems. The weighting parameters of the fuzzy-neural controller can be successfully tuned instantaneously via the genetic algorithm with a backstepping-based evaluation mechanism, instead of solving complicated mathematical equations. Using the backstepping-based evaluation mechanism can evaluate the real-time stability of the closed-loop systems in order to generate the appropriate control strategy. The simulation and experiment results show that the genetic adaptive backstepping fuzzy-neural control scheme performs on-line tracking successfully.

Acknowledgments

This work was supported by National Science Council of Taiwan, R.O.C., under Grants NSC 96-2221-E-003-012-MY3.

References

[1] Wang, L. X. "A Supervisory Controller for Fuzzy Control Systems that Guarantees Stability," *IEEE Trans. On Automatic Control*, **39**(9), 1845-1847, 1994.

[2] Leu, Y. G., Lee, T. T. and Wang, W. Y. "Observer-based Adaptive Fuzzy-Neural Control for Unknown Nonlinear Dynamical Systems," *IEEE Trans. Syst. Man, Cyber. Part B: Cybernetics*, **29**(5), 583-591, 1999.

[3] Wang, C. H., Liu, H. L. and Lin, T. C. "Direct Adaptive Fuzzy-neural Control with State Observer and Supervisory Controller for Unknown Nonlinear Dynamical Systems," *IEEE Transactions on Fuzzy Systems,* **10**(1), 39-49, 2002.

[4] Hornik, K., Stinchcombe, M. and White, H. "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, 2, 359-366, 1989.

[5] Wang, Li-Xin, *Adaptive Fuzzy Systems and Control*, Prentice Hall, 1994.

[6] Lam, H. K., Leung, F. H. and Tam, P. K. S. "Design and Stability Analysis of Fuzzy Model-based Nonlinear Controller for Nonlinear Systems using Genetic Algorithm," *IEEE Trans. Syst., Man Cybern. Part B*, **33**(2), 250–257, 2003.

[7] Chou, C. H. "Genetic Algorithm-based Optimal Fuzzy Controller Design in the Linguistic Space**,**" *IEEE Transactions on Fuzzy Systems,* **14**(3), 372-385, 2006.

[8] Cai, L., Rad, A. B. and Chan, W. L. "A Genetic Fuzzy Controller for Vehicle Automatic Steering Control," *IEEE Transactions on Vehicular Technology*, **56**(2), 529-543, 2007.

[9] Lin, F. J., Huang, P. K. and Chou, W. D. "Recurrent-Fuzzy-Neural- Network -Controlled Linear Induction Motor Servo Drive Using Genetic Algorithms**,**" *IEEE Transactions on Industrial Electronics*, **54**(3), 1449-1461, 2007.

[10] Yang, Z., Hachino, T. and Tsuji, T. "Model Reduction with Time Delay Combining the Least-squares Method with the Genetic Algorithm," *IEE Proc. Control Theory Appl.,* **143**(3), 1996.

[11] Michalewicz, Z. *Genetic Algorithms + Data Structure = Evolution Program*, Springer-Verlag, 1996.

[12] Goldberg, D. *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Addison-Wesley, 1989.

[13] Lawrence, D. *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[14] Iwasaki, M. Miwa, M. and Matsui, N. "GA-based Evolutionary Identification Algorithm for Unknown Structured Mechatronic Systems," *IEEE Transaction on Industrial Electronics,* **52**, 300-305, 2005.

[15] Gonzalez, A. and Perez, R."Selection of Relevant Features in a Fuzzy Genetic Learning Algorithm," *IEEE Transactions on Systems*, Man, and Cybernetics-Part B: Cybernetics, **31**(3), 2001.

[16] Caponetto, R., Fortuna, L., Graziani, S. and Xibilia, M. "Genetic algorithms and Applications in System Engineering: A Survey," *Trans. Inst. Meas. Control,* **15**, 143-156, 1993.

[17] Merz, P. and Freisleben, B. "Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem," *IEEE Trans. On Evolutionary Computation,* **4**(4), 337-352, 2000.

[18] Game, G. W. and James, C. D. "The Application of Genetic Algorithms to the Optimal Selection of Parameter Values in Neural Networks for Attitude Control System," *IEEE Colloquium on High Accuracy Platform Control in Space,* pp. 3/1-3/3, 1993.

[19] Moon, S. W. and Kong, S.G. **"**Block-based Neural Networks," *IEEE Transactions on Neural Networks*, **12**(2), 307 –317, 2001.

[20] Lin, C. T. and Jou, C.P." GA-based Fuzzy Reinforcement Learning for Control of a Magnetic Bearing System," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **30**(2), 276 –289, 2000.

[21] Wang, C. H., Liu, H. L. and Lin, C. T. "Dynamic Optimal Learning Rates of a Certain Class of Fuzzy Neural Networks and its Applications with Genetic Algorithm," *IEEE Transactions on Systems, Man and Cybernetics, Part B,* **31**(3), 467 –475, 2001.

[22] Farag, W. A., Quintana,V. H. and Lambert-Torres, G.” A Genetic-Based Neuro-Fuzzy Approach for Modeling and Control of Dynamical Systems,” *IEEE Transactions on Neural Networks,* **9**(5), 1998.

[23] Wang, W. Y. and Li, Y. H. “Evolutionary Learning of BMF Fuzzy-neural Networks using a Reduced-form Genetic Algorithm,” *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **33**, 966 – 976, 2003.

[24] Krstic, M., Kanellakopoulos, I. and Kokotovic, P.V. *Nonlinear and Adaptive Control Design*. New York: Wiley, 1995.

[25] Slotine,J. E. and Li, W. *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1991.

[26] Tang,Y., Tomizuka, M., Guerrero, G. and Montemayor, G. “Decentralized Robust Control of Mechanical Systems,” *IEEE Trans. Automat.Contr.*, **45**, 771-776, 2000.